Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. 6, No. 1, pp. 29-43, 2008

THE PROPOSAL OF TWO BIO-INSPIRED ALGORITHMS FOR TEXT CLUSTERING

Ana Karina F. Prior[1,2] Leandro Nunes de Castro[1,2] Leandro R. de Freitas[1] Alexandre Szabo[2]

[1]NatComp - From Nature to Business, Rua do Comércio 44, sala 03, Centro, Santos - SP – Brazil.
[2]Mackenzie University, Rua da Consolação, 896, São Paulo – SP - Brazil.
Emails: [anakarina.prior, leandrorubim ,alexandreszabo]@gmail.com, lnunes@mackenzie.br

Abstract – The Internet can be seen as a major repository of resources and information. The growing demand for information, along with the large amount of data available, has been stimulating the research of methods for text mining. This work aims at using feature selection and text clustering techniques based on a Particle Swarm Clustering (PSC) algorithm and on an Artificial Neural Network modeled as a competitive and constructive Antibody Network, called RABNET (Real-valued Antibody Network), to show that both techniques present relevant results when applied to text clustering problems.

Keywords: Text Mining, Text Clustering, PSC, RABNET, Artificial Immune Systems, Swarm Intelligence.

1 Introduction

Text mining (Weiss et al., 2005), (Hotho et al., 2005) is a field of research within the data mining (Han et al., 2000) area that has been receiving a great deal of attention over the past years. This is mainly due to the growing need to automatically analyze text data, including data available in the Internet, since the overload of text information hinders its manual analysis, location and access. Text mining can be understood as the process of extracting interesting, non-trivial and useful patterns (knowledge) from text documents (Weiss et al., 2005). Several important classes of problems can be solved using text mining, such as document classification, clustering, and information retrieval.

The main objective of this work is to apply both an artificial immune network algorithm, named RABNET (*Real-valued Antibody Network*) (Knidel et al., 2005), and a swarm intelligence algorithm, named PSC (*Particle Swarm Clustering*) (Cohen & Castro, 2006) so that they can be applied to cluster text data. In order to assess the performance of both algorithms, they were implemented and applied to cluster three benchmark text corpora. The performance measures used to evaluate and compare the algorithms were Entropy and Purity (Zhao & Karypis, 2004) and the algorithm used for benchmarking was the well known *k*-means clustering (Chakrabarti, 2003).

This paper is organized as follows. Section 2 provides a brief overview of bio-inspired computing and Section 3 briefly describes related works. Section 4 introduces the PSC algorithm and Section 5 introduces the RABNET algorithm. Section 6 provides a brief description of the similarity measure used. Section 7 provides information about the datasets used for assessment, a discussion about the sensitivity of the algorithms to its tunable parameters, the experiments performed, followed by a discussion of the results obtained. The paper is concluded in Section 8.

2 Bio-Inspired Computing

For years, many computational models, studied by scientists and engineers, brought huge benefits and solutions to humanity. Despite these technological advances, solutions to well known and complex problems, such as autonomous navigation and knowledge discovery from large databases were still scarce or inadequate. Many problems remained unsolved or poorly solved, opening the field for research of innovative solutions. Bio-inspired Computing is one of the three categories of the broad field of Natural Computing (de Castro, 2006), which emerged from the idea of using nature's inspiration to search for new computational solutions to complex problems. It is based on living organisms, their processes and behaviors evolved over thousands of years, such as self-organization, mechanisms of survival and adaptation. Inspired by nature, researchers see the possibility of creating computational models based on biological processes and phenomena. Natural computing thus involves the extraction of ideas from nature to the design of nature-inspired algorithms for solving complex problems. This field of research can be divided into three categories (de Castro, 2007):

1. Computing inspired by nature: using nature as inspiration for the development of techniques (tools) for solving complex problems. This branch involves techniques such as Neural Networks, Evolutional Computing, Swarm Intelligence and Artificial Immune Systems.

2. Simulation and emulation of nature by means of computing: involves computing mechanisms to synthesize natural behaviors, patterns and biological processes. The main lines of work are studies on life and artificial organisms, called Artificial Life, and Fractal Geometry

3. Computing with natural materials: novel computing paradigms that can lead to new generations of computers, such as DNA and quantum computing.

The objective of computing inspired by nature, or bio-inspired computing, is to use biological phenomena as a source of inspiration for the development of computational solutions to complex problems. From behavioral studies of natural systems, mathematicians, engineers and computer scientists realized that the inspiration in natural phenomena to develop computer systems can promote the development and major improvements in tools for solving various problems in different areas, such as computer science, biology, chemistry, finance, among others. Two specific bio-inspired computing areas are discussed in this paper, namely, *swarm intelligence* and *artificial immune systems*. Swarm intelligence can be subdivided into two branches. The first branch is based on the collective behavior of insects, used in solutions for problems of combinatorial optimization, clustering, and collective robotics, among others (Bonabeau et al., 1999). The second branch focuses on socio-cognitive algorithms, and has been successfully applied to perform search in continuous spaces (Kennedy et al., 2001). The research area of artificial immune systems was created by borrowing ideas from the vertebrate immune system (de Castro & Timmis, 2002).

The motivation to study bio-inspired algorithms is based on three pillars: *i*) there is a great need in seeking innovative bio-inspired solutions to complex problems; *ii*) the methods inspired by nature have shown satisfactory performance and in many cases, the best performance for many problems; *iii*) many real-world problems cannot be resolved, or are computationally intractable, by traditional methods, such as mathematical programming.

2.1 Swarm Intelligence

The term *swarm intelligence* (SI) was coined in the late 1980s to refer to cellular robotic systems in which a collection of simple agents in an environment interact according to local rules (Beni, 1988; Beni and Wang, 1989). Two main lines of research can be identified within swarm intelligence: *i*) the works based on social insects; and *ii*) the works based on the ability of human societies to process knowledge. Although the resultant approaches are quite different in sequence of steps and sources of inspiration, they present some commonalities. In general terms, both of them rely upon a population (colony or swarm) of individuals (social insects or particles) capable of interacting (directly or indirectly) with the environment and one another.

One of the algorithms to be discussed here, namely the Particle Swarm Clustering (PSC) algorithm, was designed based on the Particle Swarm (PS) Optimization algorithm (Kennedy and Eberhart, 1995). The PS algorithm has as one of its motivations to create a simulation of human social behavior; that is, the ability of human societies to process knowledge (Kennedy and Eberhart, 1995; Kennedy, 1997; Kennedy, 2004). PS takes into account a population of individuals capable of interacting with the environment and one another, in particular some of its neighbors. Thus, population level behaviors will emerge from individual interactions. Although the original approach has also been inspired by particle systems and the collective behavior of some animal societies, the main focus of the algorithm is on its social adaptation of knowledge; the same focus taken here.

In the PS algorithm, individuals searching for solutions to a given problem learn from their own past experience and from the experiences of others. Individuals evaluate themselves, compare to their neighbors and imitate only those neighbors who are superior to themselves. Therefore, individuals are able to evaluate, compare and imitate a number of possible situations the environment offers them. The most typical PS algorithm searches for optima in an L-dimensional real-valued space, $\Re^L$. Thus, the variables of a function to be optimized can be conceptualized as a vector that corresponds to a point in a multidimensional search space. Multiple individuals can thus be plotted within a single set of coordinates, where a number of individuals will correspond to a set of points or particles in the space.

2.2 Artificial Immune Systems

The line of research called *artificial immune systems* (AIS) has proven effective in the late 1990s with the publication of the first volume edited exclusively in the area (Dasgupta, 1999). From that point, several researchers from mathematics, engineering, computing and biology, which previously have developed work based on inspiration from immunology, are dedicated to the development of this new bio-inspired computational approach.

The number of applications and algorithms present in the AIS literature is vast, but some core ideas have been broadly explored, namely, *clonal selection* and *affinity maturation*, *negative selection*, and *immune networks* (de Castro & Timmis, 2002). Several new proposals involving mainly concepts from innate immunity and the danger theory have appeared in the last few years (de Castro, 2007), but still not with a common ground among them.

To design artificial immune systems, de Castro and Timmis (2002) have proposed a layered approach based on the *immune engineering* framework introduced by de Castro (2003). The immune engineering process that leads to a framework to design AIS is composed of the following basic elements (de Castro and Timmis, 2002): 1) a *representation* for the components

of the system; 2) a set of *mechanisms to evaluate the interaction* of individuals with the environment and each other. The environment is usually simulated by a set of input stimuli or patterns, one or more fitness function(s), or other means; and 3) *procedures of adaptation* that govern the dynamics and metadynamics of the system, i.e., how its behavior varies over time. The present paper presents and adapts an artificial immune system hybridized with a neural network to solve clustering problems, as will be discussed further in the text.

3 Related Works

Although there are many text mining works in the literature, few involve the use of bio-inspired algorithms emphasizing immune-inspired and swarm intelligence methods for document clustering, as will be briefly reviewed here.

In (Nanas et al., 2006), the authors proposed a self-organizing immune-inspired filter, known as Nootropia, to evaluate documents based on the users' change of interest. An immune network was used to build a network of terms that represent the user's interests and their goal was to perform information filtering and grouping so as to provide the users with only relevant information. The filter had to be able to adapt itself to the multiple interests and his/her possible change of interests. Experiments were performed using TREC-2001 (Voorhees & Harman, 2001) that adopts the Reuters Corpus Volume 1 (RCV1) (Lewis et al., 2004) and the evaluation metric used was the Average Uninterpolated Precision (AUP).

In (Secker et al., 2003) the authors proposed an immune algorithm inspired by clonal selection and cell death mechanisms for e-mail classification. The immune-inspired algorithm, called AISEC, was capable of identifying e-mails continuously as either interesting or non-interesting. Experiments were performed using a standard e-mail dataset with 2,268 e-mails of which 742 were spam and 1,526 were regular messages. The evaluation metrics used were Accuracy, Recall and Precision. They used a naïve Bayes classifier to compare the performance of their algorithm with.

With the same goal as the work mentioned above, in (Bezerra et al., 2006) the authors proposed a spam filtering approach, consisting of an antibody network, known as SRABNET, which is a variation of RABNET (Knidel et al., 2005) with supervised learning. Experiments were performed using the PU1 corpus that consists of 1,099 messages of which 481 were spam and 618 were not. The evaluation metrics used were Recall, Precision, Weighted Accuracy measure (WAcc) and Total Cost Ratio (TCR). They used a naïve Bayes classifier to compare the performance of their algorithm with.

The work presented in (Tang & Vemuri, 2005) involves the application of an artificial immune network, named aiNet, to document clustering. The authors added the k-means algorithm as another choice for clustering and also used Principal Component Analysis to reduce the dimension of the original vectors. Experiments were performed using the 20 newsgroup dataset and the used evaluation measurements were Accuracy and the F-Measure.

There are many studies in the literature involving adjustments or hybrid solutions with the PSO (Particle Swarm Optimization) algorithm for clustering problems. The work of (Omran et al., 2005) is a dynamic clustering solution, called DCPSO (Dynamic Clustering using Particle Swarm Optimization), which uses the PSO on non-supervised images classification. The main objective of this work is to automatically generate a great number of groups and simultaneously cluster the data with the minimum user's interference. Initially, the algorithm divides the data set into a large number of groups to reduce the effects of the initial conditions. Then the best number of groups is chosen through the binary PSO and the centers of these groups are refined with the *k*-means algorithm. The evaluation tests were based on six pictures and in the comparison with two well famous techniques: Kohonen's self-organizing maps and a targeting of images' color based on a grouping technique (snob). The results were quantified by index of validity described in (Omran et al., 2005). The authors concluded that the algorithm DCPSO presented a better performance than the other algorithms, and found with success a great number of groups.

In the research done by Cui in 2005, a new approach was described using the PSO algorithm to clustering tasks with the aim of finding appropriate group centroids to minimize the intra-group distance and maximize the inter-group distance. To perform this task, the basic PSO algorithm considers each particle as a possible grouping solution. Each particle maintains a matrix $\mathbf{X} = (C_1, C_2, \ldots, C_i, \ldots, C_k)$ where C_i represents the *i*-th centroid vector of the group, and *k* represents the number of groups. To evaluate the solution represented by each particle, the authors used a fitness value generated from the Equation (1), which calculates the average distance between the documents and the group's centroid.

$$f = \frac{\sum_{i=1}^{k} \left\{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \right\}}{k} \quad \textbf{(1)}$$

where m_{ij} denotes the j-th document vector that belongs to group i; o_i is the centroid vector of the i-th group; $d(o_i, m_{ij})$ is the distance between the document m_{ij} and the centroid of group i; p_i represents the number of documents that belong to C_i; and k represents the number of groups.

The authors suggested a hybrid approach of the PSO with the use of the *k*-means algorithm. This model suggested the use of the PSO results to set the number of centroids for the *k*-means algorithm. To evaluate the approach performance, the authors used four different datasets (each containing from 204 to 878 documents) with a minimum of 5000 words for each dataset. To calculate the documents similarity the authors used the cosine measure and the Euclidean distance. To evaluate the solution represented by each particle it was used the Equation (1), described previously. The authors concluded that the experiments with the hybrid algorithm (PSO and *k*-means) generated more compact and qualitative groups than when the algorithms were used separately.

4 PSC: The Particle Swarm Clustering Algorithm

The Particle Swarm Clustering (PSC) algorithm (Algorithm 1) is a method based on natural computing developed by (Cohen & de Castro, 2006) to perform data analysis. The PSC is an algorithm based on the method called Particle Swarm Optimization (PSO), proposed by (Eberhart & Kennedy, 1995) and inspired by the social adaptation of knowledge and the principle that a population of individuals is able to interact with the environment and each other, especially with its neighbors, resulting in a sort of collective intelligence.

Cohen and de Castro (2006) proposed some modifications in the PSO algorithm with the goal of developing a tool to solve clustering problems. The main modification proposed was the creation of a set of particles in the space of input data, so they become prototypes of natural groups of the input data, unlike the PSO in which each particle is a potential solution to a given problem. In order for this process to occur within a socio-cognitive perspective, each input datum must be submitted to the swarm of particles and the particle with the greater similarity to this datum will be moved towards it influenced by its best position (local experience or *cognitive term*), the position of the particle that was closer to that datum so far (global experience or *social term*) and the new added term in the proposal, called *self-organizing term*. This new term is associated with the procedures of moving the prototypes in the space toward the input datum, an approach commonly used in self-organizing systems (Kohonen, 2000). The particles' velocity is updated according to the following equation:

$$\mathbf{v}_i(t+1) = \omega.\mathbf{v}_i(t) + \varphi_1 \otimes (\mathbf{p}_i^j(t) - \mathbf{x}_i(t)) + \varphi_2 \otimes (\mathbf{g}^j(t) - \mathbf{x}_i(t)) + \varphi_3 \otimes (\mathbf{y}^j - \mathbf{x}_i(t)) \quad \textbf{(2)}$$

where ω is the inertia term that controls the particle convergence, $\mathbf{p}_i^j(t)$ is the vector containing the best position in the history of particle i in relation to the input datum j; $\mathbf{g}^j(t)$ is the vector containing the position of the best particle so far in relation to the input datum j; and $\mathbf{y}^j$ is the position of object (input datum) j.

The particles' position is updated according to the following equation:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad \textbf{(3)}$$

This equation also presents the random vectors $\boldsymbol{\varphi}_1$, $\boldsymbol{\varphi}_2$ and $\boldsymbol{\varphi}_3$ exercising influence over the cognitive, social and self-organizing terms, respectively. $\boldsymbol{\varphi}_1$ is the vector of stochastic weights for the best individual position, $\boldsymbol{\varphi}_2$ is the vector of stochastic weights for the best overall position and $\boldsymbol{\varphi}_3$ is the stochastic weights vector of the particle's distance in relation to the object.

It is important to observe that this algorithm is completely unsupervised; that is, no explicit fitness measure is employed. Particles are moved in the space based only on their similarity with the input datum. At each iteration the following algorithm is executed:

1. For each input datum, a winning particle i is selected (the one with maximum similarity with the data entry) among all the particles of the swarm. The degree of similarity is calculated by the cosine measure (Eq.(5)).
2. This particle is used to update the social term (based on the position of the particle that was closer to that datum so far), cognitive term (based on its history), self-organizing term (based on the self-organizing principle).
3. It is checked whether there are particles that did not win during the iteration. The velocity of these particles will be updated in the direction of the particle that wins the most ($\mathbf{x}_{most\_win}$) at this iteration. This is done to avoid that some particles become stagnated.
4. Update the inertia term and return to Step 1 until a maximum number of iterations, *max_it*, is reached.

```
Procedure [X] = PSC (data_set, max_it, n_part, ω, φ1, φ2, φ3, φ4, v_max, v_min)
      y = data_set
      Initialize X // usually every particle is initialized at random
      Initialize v_i // at random, v_i ∈ [v_min , v_max]
      Initialize dist_i
      t ← 1
      while t < max_it do,
```

```
            // Calculate similary between the documents and the particles
            For j = 1 to N do, // for each data
                  For i = 1 to n_part do,
                        dist_i(j) = similarity(y^j, x_j)
                  End For
                  I ← max(dist) // winning particle is selected (the one with maximum
                                // similarity with the data entry)
                  If f(x_I) < f(p_I) then
                        p_I = x_I
                  End if
                  If f(x_I) < f(g^j) then
                        g^j = x_I
                  End if
                  v_i(t+1) = ω.v_i(t)+φ1⊗(p_i^j(t)−x_i(t))+φ2⊗(g^j(t)−x_i(t))+φ3⊗(y_j−x_i(t))
                  v_i ∈ [v_min , v_max ]
                  x_i(t + 1) = x_i(t) + v_i(t + 1)
            End for
            For i = 1 to n_part do,
                  If (x_i != win) then
                        v_i(t+1) = ω.v_i(t)+φ4⊗(x_most_win−x_i(t))  v_i ∈ [v_min , v_max ]
                        x_i(t + 1) = x_i(t) + v_i(t + 1)
                  End if
            End for
            ω = 0.95 * ω
            t ← t + 1
      End while
End procedure
```

Algorithm 1: Pseudocode of the PSC algorithm

5 RABNET: A Real-Valued Antibody Network

Inspired by ideas from immunology, RABNET (Real-valued Antibody Network) (Algorithm 2) is an artificial neural network modeled as a competitive and constructive antibody network and usually applied for data clustering (Knidel et al., 2005). In order to adapt to the input patterns in a self-organizing manner, RABNET makes use of some features from an immune response, such as the clonal expansion of the most stimulated cells, the affinity maturation of the repertoire, and the death of the non-stimulated cells (de Castro & Timmis, 2002).

Unlike most clustering algorithms, RABNET does not require a pre-definition of the number of clusters to be found in a dataset. The number of antibodies in the network will be determined dynamically based on immune principles. The RABNET algorithm assumes an antigen population (**Ag**) to be recognized by an antibody repertoire (**Ab**). The RABNET algorithm can be divided into five distinct phases, as detailed in the following.

At the beginning of the adaptation process, RABNET contains a single antibody (neuron) in the network and grows when required. This single antibody is initialized randomly.

During the immune system evolution, an organism can meet a certain antigen several times. In the clustering problem to be solved, each document corresponds to one antigen (input pattern) and they are iteratively and randomly presented to the antibody network at each iteration.

5.1 Competitive Phase

The competitive phase consists of finding the most similar antibody to the antigen presented, i.e., to find the winner antibody to a given input pattern. This antibody is said to have the highest affinity with the antigen. Since we are working with text documents, the affinity level of an antibody in relation to the antigen presented will be measured by the cosine similarity (Weiss et al., 2005), to be presented further.

5.2 Network Growing

Network growing is inspired by the clonal expansion of most of the stimulated immune cells in an immune response (de Castro & Timmis, 2002). The selection of the most stimulated cell is based on the affinity to the antigen, determined during the competitive phase, and also on the concentration of antigens recognized by an antibody. Basically, the most stimulated cell (in RABNET a cell is the same as an antibody) in the immune repertoire is selected for cloning (splitting). The stimulation level of

an antibody is determined by two parameters: 1) the affinity level of antibody *j* in relation to the antigen presented; and 2) the concentration (number), (τ), of antigens recognized by antibody *j*, which is determined in the competition phase.

If the current iteration is a multiple of β:

- Antibody *I* that recognizes the highest concentration of antigens is selected (If two antibodies have the same concentration, one of them is selected randomly).
- Among all antigens (documents) recognized by antibody *I*, $\mathbf{Ag}_i$ with the lowest similarity to $\mathbf{Ab}_I$ is selected.
- If the similarity between the selected antibody and $\mathbf{Ag}_i$ is lower than a pre-defined threshold (ε), then $\mathbf{Ab}_I$ is cloned.

The weight vector of the newly created antibody receives the antigen's attributes (input pattern) with the lowest affinity to $\mathbf{Ab}_I$, that is, the one with the lowest cosine similarity to the antibody selected for cloning.

5.3 Network Pruning

The strategy adopted by the prune network antibodies is based on the concentration level of each antibody, determined in the competitive phase. If the concentration level of an antibody is zero, it means that this antibody was not stimulated by any of the antigens and can be pruned.

5.4 Weight Updating

Updating the attribute vectors of antibodies in RABNET is similar to the weights updating procedure used in competitive neural networks (Han et al., 2000). The next equation shows the weight updating rule, where α is the learning rate, and $\mathbf{Ab}_k$ the antibody that recognizes the antigen:

$$\mathbf{Ab}_k = \mathbf{Ab}_k + \alpha(t) * (\mathbf{Ag} - \mathbf{Ab}_k). \quad (4)$$

In this case, the antibodies are constantly moving in the direction of the recognized antigens. After γ iterations, the learning rate is reduced geometrically by a constant value σ: $0 \leq \alpha(0) \leq 1$.

5.5 Convergence Criterion

The convergence criterion used checks the stability of the number of neurons in the network and the variation in the weight vectors (antibodies). A parameter β controls the network growth, in which growing is tested every β iterations. It is assumed that the network topology has reached stability if during windows of $10*\beta$ iterations there is no variation in the number of neurons. Maintaining the same topology during $10*\beta$ iterations means that the structure of the network has not changed for 10 opportunities of growing and $10*\beta$ possibilities for pruning, as the pruning process is executed every iteration.

Concerning the antibodies, they are assumed to have stabilized if the sum of their modules does not vary by more than 10^{-4} from the current iteration to the past $10*\beta$ iterations.

The RABNET pseudocode is presented in Algorithm 2.

```
Procedure [] = RABNET (α, β, γ, σ, ε)
       Initialize randomly a single antibody in the network
       t ← 1
       while the convergence criterion is not reached do,
              For each input pattern (antigen) do,
                     Present a random antigen to the network
                     Calculate similarity between the antigen and the antibodies in the network
                     Find the winning antibody
                     Increase the concentration level of the winner
                     Update the weights of the winner antibody (Eq. 4)
              End for
              If iteration > γ then
                     α = σ * α
              End if
              If iteration is multiple of β then
                     Grow if necessary
              End if
              If the concentration level of a given antibody is zero then
                     Prune it from the network
              End if
              t ← t + 1
```

```
        End while
End procedure
```

Algorithm 2: Pseudocode of the RABNET algorithm

6 Document Clustering with RABNET and PSC

In clustering algorithms, a set of data to be clustered is represented as a set of vectors $\mathbf{z} = (z_1, z_2, ..., z_m)$, in which z_j corresponds to a single object called feature vector. The array of features includes suitable characteristics to represent the object. To adapt the algorithm to cluster text data, each object is considered as a text document and all objects are represented by the Vector Space Model (VSM) (Salton et al., 1975). In this model, the content of a document is formalized as a point in a multidimensional space and represented by a vector $\mathbf{u}$, where $\mathbf{u} = \{w_1 , w_{2,} ..., w_n\}$ and w_i ($i = 1, 2, ..., n$) is the term weight in a document. In both algorithms, the weight of the term is determined according to the tf-idf measure which calculates the significance or frequency of the term within the document and for all documents. To compute the similarity between the documents (weight vectors) and the prototypes, the algorithms apply the Cosine similarity measure:

$$\text{similarity}(\mathbf{z}_a,\mathbf{z}) = \cos(\mathbf{z}_a,\mathbf{z}) = (\mathbf{z}_a \cdot \mathbf{z})/(\|\mathbf{z}_a\|_2 * \|\mathbf{z}\|_2) \quad (5)$$

where $\mathbf{z}_a$ and $\mathbf{z}$ are the vectors, $\| \mathbf{z} \|_2$ is the Euclidean norm, and $\mathbf{z}_a \cdot \mathbf{z}$ corresponds to the inner product between vectors $\mathbf{z}_a$ and $\mathbf{z}$.

7 Performance Assessment

In order to assess the performance of the algorithms in the context of text clustering, both algorithms were applied to three different benchmark datasets, and their results compared to that of the *k*-means algorithm. As RABNET automatically determines the number of antibodies (neurons) to be used in the network, it was run first and, then, the same number of units determined by RABNET was adopted as *k* for the *k*-means and the number of particles for the PSC.

7.1 Materials and Methods

Text mining is a multidisciplinary field of investigation that requires background knowledge from areas such as Information Retrieval (Baeza-Yates & Ribeiro-Neto, 1999), Statistics, and Linguistics (Weiss et al., 2005), (Hotho et al., 2005). Text Mining aims at extracting regularities and patterns in large volumes of natural language texts, usually with specific objectives (Weiss et al., 2005), (Hotho et al., 2005). The following steps in text mining were used in this work: 1) tokenization; 2) dictionary generation, including stopwords removal, stemming, vector generation and storage; 3) feature selection by information gain; 4) documents comparison; and 5) clustering.

Considering recommender systems, the most used measures to evaluate the results are *precision* and *recall*, common tasks in information retrieval (Baeza-Yates & Ribeiro-Neto, 1999). The main difficulty in using this metrics in information retrieval is that the relevance of an item is inherently subjective and this can only be better evaluated knowing the profile of each user. For the particular case of clustering, metrics such as *Entropy* and *Purity* are widely used in literature (Zhao & Karypis, 2004). Therefore, in this paper we used the Entropy (E) and Purity (P) measures to assess the performance of the algorithms. This is possible since that the groups are known in advance, but this information is used here only for benchmarking purposes, which is a common practice in the literature (Crabtree et al., 2005), (Zhao and Karypis, 2004). Given a cluster S_r of size n_r, the entropy $E(S_r)$ of this cluster can be measured as follows:

$$E(S_r) = -\frac{1}{\log c}\sum_{i=1}^{k}\frac{n_r^i}{n_r}\log\frac{n_r^i}{n_r} \cdot \quad (6)$$

where c is the number of classes in the set of documents, and n_r^i is the number of documents of class i in cluster S_r. The global entropy can then be calculated as the sum of the entropies obtained in each cluster, weighted by the size of each cluster:

$$E_{global} = \sum_{r=1}^{c}\frac{n_r}{n}E(S_r) \cdot \quad (7)$$

Purity provides the ratio of the dominant class in the cluster in relation to the size of the cluster itself, and can be calculated as follows:

$$P(S_r) = \frac{1}{n_r}\max_i(n_r^i) \cdot \quad (8)$$

The global purity can be obtained by:

$$P_{global} = \sum_{r=1}^{c} \frac{n_r}{n} P(S_r) \cdot \quad (9)$$

where n_r is defined as above.

7.2 Datasets

To assess the performance of RABNET, PSC and *k*-means, three public text datasets were chosen. The first one is the Reuters-21,578 dataset (Lewis et al., 2004) compiled by David Lewis and originally collected by the Carnegie Group from the Reuters newswire in 1987. This data is usually employed by taking documents from the most frequent classes, for instance, from the 115, 90, or 10 most frequent categories (Sebastiani, 2002), (Joachims, 1998). In the experiments reported here we took the texts from the 10 most frequent categories, namely earn, acquisition, money-fx, grain, crude, trade, interest, ship, wheat and corn. This leads to a total of 7,193 documents.

The second text collection used was taken from Ohsumed (Hersh et al., 1994), compiled by William Hersh. This collection is based on a total of 50,216 medical abstracts of cardiovascular diseases from 1991. The grouping task considered here is to assign the documents to one category of the 10 most frequent diseases categories (C01, C04, C06, C08, C10, C12, C14, C20, C21, C23), what leads to a total of 7,878 documents.

To assess the scaling performance of the algorithms; that is, performance in relation to an increase in the text dataset, both collections were individually divided into three subsamples with the 500, 1,000 and 2,000 most relevant attributes of each dataset, based on their ranked information gain. Each subsample contains the number of categories of the collection taken for experimentation here; that is, 10 categories (Table 1). The documents were processed using the methods listed previously.

Table 1. Number of documents in the ten largest classes of the Reuters (RE) and Ohsumed (OH) datasets, the total number of documents (TND) used in each subsample, and the number of tokens in each dictionary generated (NT).

Category	500		1000		2000		5000		6984	
	RE	**OH**	**RE**	**OH**	**RE**	**OH**	**RE**	**OH**	**RE**	**OH**
1	1649	423	1649	423	1649	423	1650	423	1650	423
2	181	1163	181	1163	181	1163	181	1163	181	1163
3	389	588	389	588	389	588	389	588	389	588
4	2877	473	2877	473	2877	473	2877	473	2877	473
5	433	619	433	621	433	621	433	621	433	621
6	347	490	347	491	347	491	347	491	347	491
7	538	1249	538	1249	538	1249	538	1249	538	1249
8	197	525	197	525	197	525	197	525	197	525
9	369	545	369	546	369	546	369	546	369	546
10	212	1798	212	1799	212	1799	212	1799	212	1799
TND	7192	7873	7192	7878	7192	7878	7193	7878	7193	7878
NT	500	500	1000	1000	2000	2000	5000	5000	6984	6984

The third text collection used in this paper was the Spambase dataset (UCI Repository of Machine Learning Databases) collected by George Forman. This collection contains 4,601 instances of e-mails with 48 attributes each, of which 1,813 are spam and 2,788 are not. The collection of spam e-mails came from the postmaster and individuals who had filed spam and the collection of non-spam e-mails came from filed work and personal e-mails. Advertisements for products/web sites, "make money fast" schemes, chain letters and pornography were considered spam. This dataset was already pre-processed.

7.3 About the Tuning Parameters

This section presents a brief discussion of how the main parameters of RABNET and PSC are chosen so that they can be applied to a problem.

7.3.1 RABNET

The application of RABNET in text mining problems requires the definition of parameters α, β, γ, σ and the affinity threshold ε. To study the influence of these parameters in the performance of RABNET, a simplified sensitivity analysis of the algorithm will be performed by applying it to the Reuters dataset with 500 attributes. Ten simulations were performed and the results to be presented show the mean and standard deviation, out of these ten executions. These simulations were run using the following parameters: $\alpha = 0.95$, $\beta = 2$, $\gamma = 100$, $\sigma = 0.95$ and $\varepsilon = 10^{-8}$, varying each parameter according to the analysis.

Table 2 shows that parameter ε influences the specificity of the network cells, and thus the final number of neurons in the network: the higher the value of ε, the larger the number of neurons in the network and vice-versa. By contrast, it shows that, by reducing ε, little influence is observed in the Entropy and Purity measures.

Table 2. Trade off between the number of cells in the network, Entropy, Purity and the threshold ε.

ε	10^{-3}	10^{-5}	10^{-8}
Number of cells	58 cells	57 cells	54 cells
Entropy	0.23±0.01	0.23±0.01	0.23±0.01
Purity	0.80±0.01	0.80±0.01	0.80±0.01

To study the influence of γ and σ in final the network size and in the number of iterations for convergence three different values of each of these parameters were adopted: $\gamma \in \{20,100,200\}$; $\sigma \in \{0.1,0.5,0.95\}$. The results are summarized in Table 3. Parameters γ and σ strongly influence the network size and number of iterations for convergence; higher values of γ and σ result in longer convergence times. Furthermore, we found that the entropy is inversely proportional to γ, while the purity does not vary significantly in relation to it. In the case of parameter σ, the entropy was improved with the increase of this parameter, and there was little gain in performance for σ varying from 0.5 to 0.95. This analysis allows us to conclude that γ can be kept low (< 100, for example, 20), while σ can be maintained at intermediate values, such as $\sigma = 0.5$. In those situations the algorithm should present a good relationship between the performance of clustering and parsimony of the solution.

Table 3. Final number of neurons, iterations, Entropy and Purity for different values of γ and σ.

	γ			σ		
	20	100	200	0.1	0.5	0.95
Neurons	20	55	96	5	38	54
Iterations	149	223	321	191	213	223
Entropy	0.24±0.01	0.23±0.01	0.22±0.004	0.33±0.02	0.24±0.01	0.23±0.01
Purity	0.80±0.02	0.80±0.01	0.80±0.00	0.70±0.02	0.81±0.01	0.80±0.01

To assess the influence of the learning rate (α) in the network size, three values were chosen: $\alpha \in \{0.2, 0.5, 0.95\}$. The results are shown in Table 4. It is possible to observe that the learning rate (α) has almost no influence in the final number of neurons.

Table 4. Final number of neurons, Entropy and Purity for different values of α

α	0.2	0.5	0.95
Neurons	51	51	55
Entropy	0.33±0.02	0.24±0.01	0.23±0.01
Purity	0.70±0.02	0.81±0.01	0.80±0.01

The parameter β controls the network growth, and strongly influence the number of iterations for convergence. Higher values of β which result in long convergence time. It was possible to observe that the parameter β has no influence in the entropy and purity measures.

This simplified analysis allows us to suggest the following configuration of parameters for the generic application of the algorithm:

- Affinity threshold between antigens and antibodies: $\varepsilon \approx 10^{-8}$;
- Initial learning rate: $0.5 \leq \alpha \leq 0.95$;
- Number of iterations from which the learning rate decreases: $20 \leq \gamma \leq 100$;
- Decreasing factor for the learning rate: $0.5 \leq \sigma \leq 0.95$.
- Network growth parameter: $\beta = 2$.

The simulations to be presented in the next section were run using the following values: $\alpha = 0.95$, $\beta = 2$, $\gamma = 100$ and σ=0.15. It was observed that ε is sensitive to the size of the database. To perform the following simulations the ε parameter was set with different values for each one of the seven datasets tested, so that the algorithm could find a ideal number of clusters that would be a value close to the number of original groups of the dataset.

7.3.2 PSC

To tune the PSC parameters, Cohen and de Castro (2006) performed a sensitivity analysis of the algorithm by varying some of them for the Ruspini (Kaufman et al., 1990) dataset. From this analysis the authors suggested the following values: *max_it* = 200, $\omega = 0.95$, $v_{max} = 0.01$, $v_{min} = -0.01$, $\varphi_1, \varphi_2 \in [0.1\ 2.05]$, $\varphi_3 \in [0.005\ 1]$ and *n_part* containing the same amount of classes as the original databases. To measure the similarity between the particles and each input datum, the Euclidean distance was used. The data for testing were downloaded from the UCI Machine Learning Repository (Merz & Murphy, 1998), and a bioinformatics dataset was also used. The parameter ω is used to control the convergence. It was observed that the parameters φ_1 (cognitive), φ_2 (social) and φ_3 (self-organizing) are crucial for the performance of the algorithm. However, the influence of the social and cognitive terms in the algorithm increases when the self-organizing term decreases, and plays an important role in avoiding stagnation. After running the algorithm several times to see which were the best values to run the algorithm, the following parameters were suggested : *max_it* = 200, $\omega = 0.09$, $v_{max} = 0.1$, $\varphi_1, \varphi_2, \varphi_3, \varphi_4 \in [0\ \ 1]$.

7.4 Experimental Results

The algorithms were implemented in Java, and the experiments run on a PC Intel Pentium 4 2.4 GHz with 512MB of RAM memory. Ten simulations were performed for each subsample and each algorithm. The results presented in the following tables show the mean and standard deviation, out of ten executions, for each dataset and algorithm.

Table 5. Mean ± standard deviation of the Entropy and Purity measures for the *k*-means, RABNET and PSC algorithms applied to the Ohsumed dataset.

Ohsumed		**500**	**1000**	**2000**
***k*-means**	**Entropy**	0.68±0.01	0.82±0.01	0.62±0.02
	Purity	0.38±0.02	0.36±0.02	0.38±0.01
	Time	06m56s	08m39s	44m30s
RABNET	**Entropy**	0.66±0.04	0.82±0.04	0.60±0.05
	Purity	0.40±0.01	0.37±0.01	0.41±0.01
	Time	1h45m32s	2h49m25s	16h21m27s
PSC	**Entropy**	0.68±0.01	0.80±0.01	0.60±0.01
	Purity	0.40±0.01	0.38±0.01	0.41±0.01
	Time	01h14m47s	01h51m51s	06h43m26s

Table 6. Mean ± standard deviation of the Entropy and Purity measures for the *k*-means, RABNET and PSC algorithms applied to the Reuters dataset.

Reuters		**500**	**1000**	**2000**
***k*-means**	**Entropy**	0.31±0.02	0.29±0.02	0.26±0.02
	Purity	0.73±0.03	0.74±0.03	0.76±0.02
	Time	04m37s	09m40s	25m03s
RABNET	**Entropy**	0.29±0.02	0.30±0.03	0.22±0.02
	Purity	0.73±0.02	0.72±0.02	0.78±0.01
	Time	1h46m31s	3h55m05s	15h17m15s
PSC	**Entropy**	0.30±0.01	0.30±0.01	0.23±0.02
	Purity	0.73±0.02	0.73±0.02	0.77±0.02
	Time	55m29s	01h50m26s	04h59m07s

Table 7. Mean ± standard deviation of the Entropy and Purity measures for the *k*-means, RABNET and PSC algorithms applied to the Spambase dataset.

***k*-means**	**Entropy**	0.48±0.03	**RABNET**	**Entropy**	0.37±0.08	**PSC**	**Entropy**	0.46±0.02
	Purity	0.69±0.04		**Purity**	0.72±0.02		**Purity**	0.74±0.03
	Time	02s		**Time**	04m48s		**Time**	1m40s

It is possible to observe that all algorithms had very similar results in terms of entropy and purity. By observing the results presented in Table 5 (Ohsumed dataset) it can be noted that the PSC algorithm performed slightly better, on average, in all simulations. The results presented in Table 6 (Reuters dataset) show that RABNET performed better for the subsamples with 500 and 2000 attributes, while the K-Means performed better for the subsample with 1000 attributes. Furthermore, all algorithms performed better in the Reuters dataset than in the Ohsumed corpora, mainly due to the nature of these data. Whilst the Reuters data is composed of news (which contain, in essence, distinct information), the Ohsumed data contains medical abstracts of the same types of diseases, possibly confusing the clustering algorithms. Considering the Ohsumed dataset, it can be noted that all algorithms had a better performance for the subsample with 2000 attributes. This may have happened because the algorithms found a higher number of groups that resulted in more homogeneous and purer groups of texts. Considering the Reuters dataset, the results also show that the algorithms are reasonably robust in relation to their performances. The algorithms showed a slight improvement for the entropy and purity when the size of the dataset increased.

Considering the Spambase dataset, the results show that the RABNET had much better performance for the entropy than the other algorithms, but the PSC algorithm performed better for the purity value.

The time results show that despite having good results, the RABNET algorithm takes much more time to execute than the PSC and K-Means algorithm.

To assess the significance of the difference in performance of the algorithms, an analysis of variance assuming normality in their results (which was observed empirically) was performed. The results showed that the null hypothesis could not be rejected, which means that for the entropy and purity measures of the Ohsumed and Reuters datasets the mean values obtained were equivalent.

7.5 Comparison with Results from the Literature

Comparing the performance of an algorithm for text grouping with the results available in literature is a difficult task for several reasons:

- Each algorithm is implemented in a different way;
- The text mining pre-processing techniques used, for example, tokenization and stemming, can be implemented in different ways, what will result in distinct attribute vectors and, usually, the details of those techniques are not described in the works that emphasize the grouping or prediction methods. It means that an individual text database can result in different numerical databases and, therefore, it will result in distinct performance of the algorithms. Exceptions for these cases are those, like the Spambase, that already provide the pre-processed data; and
- There is no standardization in the text mining literature, mainly in relation to the performance measures used. In many cases, measures of predictive accuracy are used, like the percentage of correct classification, and, in others, measures like entropy and purity of the groups formed are provided, just as used in the present work.

Despite these difficulties, this section compares the performance of the k-means, PSC and RABNET algorithms with the algorithm proposed in Zhao and Karypis (2004). In that work the authors applied specific criteria functions transforming the grouping problem into an optimization task, which is substantially different from the proposal presented here, which operates based on a fully unsupervised method. That is, the PSC and RABNET algorithms do not optimize an explicit criterion function, but adapt a self-organized set of prototypes that move in the space so as to find statistical regularities within the input data.

The Reuters and Ohsumed datasets were also used for the experiments reported in (Zhao & Karypis, 2004). Thus, the entropy and purity measures of PSC and RABNET were directly compared with the best results of the entropy and purity described in Zhao and Karypis (2004). Both works used the documents in the vector space model, performed pre-processing, like frequent words removal, stemming, the removal of words that appear in just two distinct documents, and defined weights for each dictionary's word (according to the frequency of the word in the documents in which it appears and proportionally to all documents), and used the cosine similarity measure to compare documents. Basically, The experiments may differ in the following aspects:

- Pre-processing: formatting the dictionary of frequent words and the choice of the stemming algorithms were not detailed Zhao & Karypis (2004).
- Algorithms' implementation: this work presents two new bio-inspired techniques to group documents, while the work proposed by Zhao and Karypis (2004) uses optimization algorithms to maximize or minimize the criterion functions involving, for example, similarity of objects and intra-group vs inter-group dissimilarity.
- Datasets: despite using the same datasets (Ohsumed and Reuters), the samples may be different, since this work uses information gain and specific pre-processing steps, as discussed previously.

Table 8 shows the number of documents, tokens and classes used by Zhao and Karypis (2004) and Table 9 presents the best results of entropy and purity reported in Zhao and Karypis (2004), contrasting them with the results of the k-means, RABNET and PSC algorithms. Although the experiments of Zhao and Karypis were obtained with a much lower number of documents than those used in this work, the number of classes used was higher than the ten used in the experiments reported here. Whilst RABNET showed the best performance for the Reuters dataset, the method of Zhao and Karypis (2004) had a better performance for the Ohsumed dataset.

Table 8. Distribution of documents used in the work of Zhao and Karypis (2004).

Dataset	Number of documents	Number of tokens	Number of classes
Reuters-01	1504	2886	13
Reuters-02	1657	3758	25
Ohsumed	11162	11465	10

Table 9. Best values of Entropy and Purity for a *k*-way algorithm, where $k = 20$, of Zhao & Karypis (2004) and best results for RABNET, PSC and *k*-Means.

		Zhao & Karypis	RABNET	PSC	k-Means
Reuters-01	Entropy	0.32	0.22	0.23	0.26
	Purity	0.69	0.78	0.77	0.76
Reuters-02	Entropy	0.32	0.22	0.23	0.26
	Purity	0.68	0.78	0.77	0.76

Ohsumed	Entropy	0.51	0.62	0.60	0.60
	Purity	0.62	0.38	0.41	0.41

8 Conclusion and Future Trends

The main objective of this work was to adapt the Real-Valued Antibody Network, RABNET, and the Particle Swarm Clustering algorithm, PSC, to perform text clustering and assess their performance. The original ABNET algorithm was designed to cluster binary data (de Castro et al., 2003), and later adapted, becoming RABNET, to cluster numerical data in general (Knidel et al., 2005). The adaptations performed were mainly aimed at allowing the algorithm to perform well on sparse data with a large number of real-valued attributes. The same holds true for the Particle Swarm Clustering algorithm.

The results presented here suggest that the both algorithms are suitable for the development of text grouping tools. As could be observed, the RABNET and PSC algorithms had a better performance, on average, than the *k*-means, finding more homogeneous and purer groups of texts. However, the computational complexity required to cluster the text corpora used was much higher for the bio-inspired algorithms, and this deserves further investigation. Further investigations also include their application to other databases from the literature and the comparison with other techniques. The influence of the cognitive, social and self-organizing terms in the PSC algorithm also has to be carefully investigated and possibly modifications may be introduced aiming at improving its overall performance.

9 Acknowledgements

The authors thank CNPq, Fapesp, and Mackpesquisa for the financial support. The authors also thank all the reviewers for their valuable comments and suggestions.

10 References

Alsabati, K., S. Ranka, and V. Singh, An Efficient K-Means Clustering Algorithm, Proc. First Workshop on High-Performance Data Mining, (1998).

Baeza-Yates, R. and B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, (1999).

Barrie, J. M. and D.E. Presti, Collaborative Filtering, Science, 274, pp. 371-372, (1996).

Beni, G., The Concept of Cellular Robotic Systems, Proc. of the IEEE Int. Symp. on Intelligent Control, pp. 57-62 (1988).

Beni, G. and J. Wang, Swarm Intelligence, Proc. of the 7th Annual Meeting of the Robotics Society of Japan, pp. 425-428 (1989).

Berkhin, P., A Survey of Clustering Data Mining Techniques, In J. Kogan, C. Nicholas, and M. Teboulle (Eds.), Grouping Multidimensional Data, Springer, pp. 25-71, (2006).

Bezerra, G. B., T. V. Barra, H.M. Ferreira, H. Knidel, L.N. de Castro, and F. J. Von Zuben, F. J., An Immunological Filter for Spam, Conference on Artificial Immune Systems (ICARIS 2006), LNCS, 2006 – Springer, pp. 446 – 458, (2006).

Bonabeau, E., M. Dorigo, M. and G. Theraulaz, *Swarm Intelligence from Natural to Artificial Systems*, Oxford University Press (1999).

Bhakrabarti, S., Mining the Web – Discovering Knowledge from Hypertext Data, Morgan Kaufmann, (2003).

Cohen, S. C. M, and L. N. de Castro, Data Clustering with Particle Swarms, Evolutionary Computation, CEC 206, IEEE Congress on. pp. 1792-1798. (2006).

Crabtree, D., X. Gao, and P. Andreae, Universal Evaluation Method for Web Clustering Results, Technical Report of Vicrotia University of Wellington, (2005).

Cui, X., T. E. Potok, and P. Palathingal, Document clustering using particle swarm optimization, Swarm Intelligence Symposium, SIS 2005, Proc. 2005 IEEE, pp. 185–191 (2005).

Dasgupta, D., Artificial Immune Systems and Their Applications, Springer-Verlag (1999).

de Castro, L. N., Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, Chapman & Hall/CRC, pp. 267-323 (2006).

de Castro, L. N., Fundamentals of natural computing: An overview, Physics of Life Reviews, **4**, pp. 1-36 (2007).

de Castro, L. N., and J. I. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach, Springer-Verlag, (2002)

de Castro , L. N., Immune Cognition, Micro-evolution, and a Personal Account on Immune Engineering, *S.E.E.D. Journal (Semiotics, Evolution, Energy, and Development)*, University of Toronto, **3**(3), pp.134–155 (2003).

de Castro, L. N.; Von Zuben, F. J.; de Deus Jr., G. A. The construction of a boolean competitive neural network using ideas from immunology. Neurocomputing 50, pp. 51–85, (2003).

Han, J., M. Kamber, and M. Kaufmann, Data Mining: Concepts and Techniques, Morgan Kaufmann, (2000).

Haykin, S., Neural Networks – A Comprehensive Foundation, Prentice Hall, (1999).

Hersh, W., C. Buckley, T. J. Leone and D. Hickam, OHSUMED: an interactive retrieval evaluation and new large test collection for research, Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, p.192-201, July 03-06, 1994, Dublin, Ireland, (1994)

Hotho, A., A. Nürnberger, and G. Paaß, A Brief Survey of Text Mining, GLDV-Journal for Comp. Linguistics and Language Technology, 20(1), pp. 19-62, (2005).

Joachims, T., Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Proceedings of ECML-98, 10th European Conference on Machine Learning, pp.137-142, (1998).

Kaufman, L., and P.J. Rousseew, Finding Groups in Data – An Introduction to Cluster Alaysis, Wiley Series in ability and Mathematical statistics, John Wiley & Sons Inc, (1990).

Kennedy, J., "Particle Swarms: Optimization Based on Sociocognition", In L. N. de Castro and F. J. Von Zuben, *Recent Developments in Biologically Inspired Computing*, Idea Group Publishing, Chapter X, pp. 235–269 (2004).

Kennedy, J., R. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publishers (2001).

Kennedy, J. and Eberhart, R., Particle Swarm Optimization, Proc. of the IEEE Int. Conf. on Neural Networks, Perth, Australia, **4**, pp. 1942–1948 (1995).

Kennedy, J., Thinking is Social: Experiments with the Adaptive Culture Model, Journal of Conflict Resolution, **42**, pp. 56–76 (1997).

Knidel, H., L. de Castro, and F. Von Zuben., RABNET: A Real Valued Antibody Network for Data Clustering, In: International Conference on Natural Computation, 2005, Changsha. Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol. 3610, pp. 1279-1288, (2005).

Kohonen, T., Self-Organizing Maps, Springer-Verlag (2000).

Kullback, S., and R. A. Leibler, On Information ans Sufficiency, Annals of Mathematical Statistics 22, pp: 79-86, (1951).

Lewis, D., Reuters-21578 Text Categorization Text Collection, http://dit.unitn.it/~moschitt/corpora.htm (2004).

Macqueen, J. B., Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1, pp: 281-297, (1967).

Nanas, N., A. de Roeck, and V. Uren, Immune-Inspired Adaptative Information Filtering, Proceedings of ICARIS-2006, 5th International Conference on Artificial Immune Systems, LNCS, 2006 – Springer, pp. 418-431, (2006).

Omran, M. G. H., A. P. Engelbrecht, and A. Salman, Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification, Proceedings of World Academy of Science, Engineering and Technology, Vol. 9 (2005).

Rouff, C., M. Hinchey, T. Truszkowski, and J. Rash, Formal methods for autonomic and swarm-based systems, In 1st International Symposium on Leveraging Applications of Formal Methods (ISoLA 2004), Cyprus (2005).

Salton, G., A. Wong, and C. S. Yang, A Vector Space Model for Information Retrieval, Journal of the American Society for Information Science, 18(11), pp. 613-620, (1975).

Sebastiani, F., Machine learning in automated text categorization, ACM Computing Surveys, 34(1):1–47, (2002).

Secker, A., A. Freitas, and J. Timmis, AISEC an Artificial Immune System for E-Mail Classification, Evolutionary Computation, 2003. CEC'03. The 2003 Congress on. pp. 131- 138 Vol.1 (2003).

Tang, N., and V. R. Vemuri, An Artificial Immune System Approach to Document Clustering, Proceedings of the 2005 ACM symposium on Applied computing. pp. 918 – 922. (2005).

UCI Repository of Machine Learning Databases, "On line Datasets", http://archive.ics.uci.edu/ml/datasets/Spambase

Vorhees, E. M., and D. Harman, Overview of TREC 2001, National Institute of Standards and Technology, Gaithersburg, (2001).

Weiss, S., N. Indurkhya, T. Zhang, and F. Damerau, Text Mining: Predictive Methods for Analyzing Unstructured Information, Springer (2004).

Zhao, Y., and G. Karypis, Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, Machine Learning 55(3), pp. 311-331 (2004).